INFINIBAND FOR HIGH ENERGY PHYSICS

A. Heiss, U. Schwickerath, Forschungszentrum Karlsruhe, 76021 Karlsruhe, Germany
G. Ganis, CERN

Abstract

Distributed physics analysis techniques and parallel applications require a fast and efficient interconnect between compute nodes and I/O nodes. Apart from the required bandwidth the latency of message transfers is important, in particular in environments with many nodes. Ethernet is known to have high latencies of 30 μs to 60 μs for the common gigabit Ethernet hardware. The InfiniBand architecture is a relatively new, open industry standard. It defines a switched high-speed, low-latency fabric designed to connect compute nodes and I/O nodes with copper or fibre cables. The theoretical bandwidth is up to 30 Gbit/s. The Institute for Scientific Computing (IWR) at the Forschungszentrum Karlsruhe is testing InfiniBand technology since begin of 2003 and runs a cluster of dual Xeon nodes using the 4X (10 Gbit/s) version of the interconnect. Bringing the RFIO protocol – which is part of the CERN CASTOR facilities for sequential file transfers – to InfiniBand has been a big success, allowing significant reduction of CPU utilization and increase of file transfer speed. Performance results of RFIO on Xeon and Opteron platforms are presented. To simulate a typical situation in physics analysis of most up to date data, the RFIO daemon is stress tested in a multi-user environment. A first prototype of a direct interface to InfiniBand for the ROOT toolkit is currently designed and implemented. Furthermore, experiences with hard- and software, in particular MPI performance results, will be reported.

INFINIBAND

The InfiniBand specifications[1] define a switched, high bandwidth, low latency fabric for both inter-process and I/O communication. 1, 4 or 12 wire pairs for each direction carry voltage differential signals and provide bidirectional bandwidths of 2.5 Gbit/s (1X), 10 Gbit/s (4X) or 30 Gbit/s (12X). Due to 8B/10B encoding, the maximum usable bandwidth is 250 MB/s, 1 GB/s or 3 GB/s respectively. Host Channel Adapters (HCA) connect the host bus to the InfiniBand fabric. They are available in PCI-X (133 MHz, 64 bit) and PCI-Express versions, providing two 4X-InfiniBand connectors for redundant connections. Copper cables (suitable for distances up to $\approx$15m) and fibre optic cables are available. Switches are available with up to 288 ports. Gateway modules provide connectivity with Fibre Channel and Ethernet. InfiniBand is remote DMA (RDMA) capable, allowing high transfer rates with very low CPU utilization. For "reliable connections" the hardware takes care of data integrity. The transport layer is queue based. A queue pair (QP) consists of a send and a receive queue which can be used for data transfers. Completed transfers are reported in a completion queue. The specifications also define a set of "verbs" to access the HCA. However, the detailed implementation of the verbs API is vendor specific. A set of higher level protocols is supported. IPoIB (IP over InfiniBand) allows for TCP/IP communication using virtual Ethernet interfaces on the hosts connected to the InfiniBand fabric. With SRP (SCSI Remote Protocol), InfiniBand can be used as a storage network. Several MPI (Message Passing Interface) implementations[2][3][4] are available. Since approximately begin of 2004, a reference implementation of the HCA drivers and development kit available as open source from Mellanox[5], a major InfiniBand chip vendor. Drivers are available for many platforms (IA32, IA64, X86_64, Power-PC, ...) and operating systems. Meanwhile, most InfiniBand vendors joined the OpenIB[6] project to provide improved drivers and software as open source.

HARDWARE SETUP

The Institute for Scientific Computing (IWR) runs a cluster of 13 dual-Xeon compute nodes and one interactive dual-Xeon node with 4X-InfiniBand interconnect. The Intel E7501-chipset based compute nodes (Supermicro X5DPR-iG2+ boards) are equipped with two Xeon 2.4 GHz CPUs, 2 GB of main memory and an Infinicon InfiniServ7000 HCA plugged into a 133MHz/64bit PCI-X slot. All machines are connected to a Mellanox InfiniScale MT43132 16-port switch which provides roughly 1 GB/s of bandwidth between any two ports. The operating system is RedHat Linux 7.3 based, but a 2.4.26 vanilla kernel is used. Scientific Linux CERN[7] is currently being tested.

Besides the Xeon cluster, 2-way AMD Opteron machines with 2.2 GHz Opteron 248 CPUs (Sun Fire V20z) and a 4-way Opteron 848 machine with 16 GB of main memory, each equipped with Mellanox InfiniHost HCAs ("Cougar Boards") with 133MHz/64bit PCI-X interface were used for measurements. These machines were tested with 64-bit versions of the SuSE Linux Enterprise Server 8 and Scientific Linux CERN operating systems.

PARALLEL COMPUTING

MPI over InfiniBand

As MPI implementation, the MPICH v1.2.5.2 library[8] with OSU[2] v0.9.2 patch "MVAPICH" (MPI-1 over VAPI for InfiniBand) is provided for the GCC 3.2 and Intel 7.1 compilers on the Xeon cluster. The same MPI implementation was used on the Opteron test machines. The total system performance of the Xeon cluster was measured with the HPL benchmark[9] to be almost 92 Gflops using all 26 CPUs (Fig. 1). This corresponds to ≈74% of the theoretical peak performance of 124.8 Gflops.

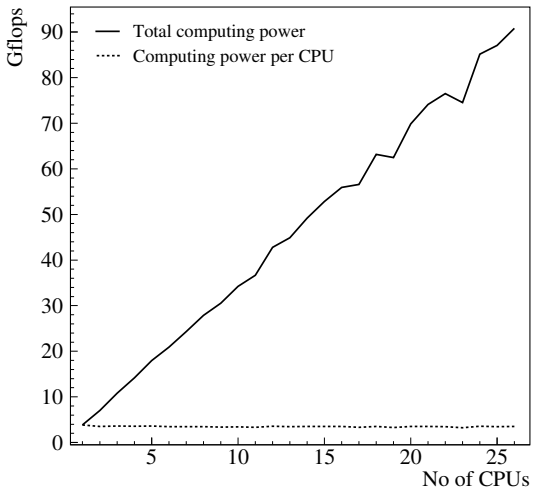

Figure 1: System performance as a function of the number of used CPUs measured with the HPL benchmark.

Bandwidth and Latency

The maximum sustained data rate that can be achieved at the network level was measured with a simple bandwidth test program available from the OSU web page[2] which uses non-blocking MPI_Isend and MPI_Irecv functions. Latency measurements were done using a ping-pong type test program which uses blocking MPI_Send and MPI_Recv functions. This program is also available from [2]. Between two of the Xeon compute nodes, the maximum bandwidth of 798 MB/s was achieved for 256 kB message size. The latency for 1-byte messages is 6.5 μs. (It is known that somewhat better performance is possible on Xeon systems using a ServerWorks chipset.) When the two communicating processes run on the same dual-Xeon node, the maximum bandwidth reaches 883 MB/s for 16k messages, but drops to approximately 400 MB/s for larger messages. The latency is 1.3 μs in this case. A first quick test with the recently released MVAPICH version 0.9.4 indicated that this problem is under control now. Between two Opteron systems, 825 MB/s peak bandwidth was achieved and the 1-byte latency was measured to be 5.5 μs. The bandwidth between processes running on the same dual-Opteron machine reaches 655 MB/s for 8k message size and drops to roughly 400 MB/s and lower for larger messages. With the new 0.9.4 version of MVAPICH however, a SMP bandwidth of up to 1 GB/s was achieved. The 1-byte latency is about 0.5 μs.

A Lattice–QCD program[10], parallelized in two dimensions, was run on the Xeon-cluster. This very memory and communication intensive application suffers from the bottleneck caused by the single memory bus in the dual-Xeon systems. If only one CPU per node is used, the speedup is clearly better (Fig. 2).

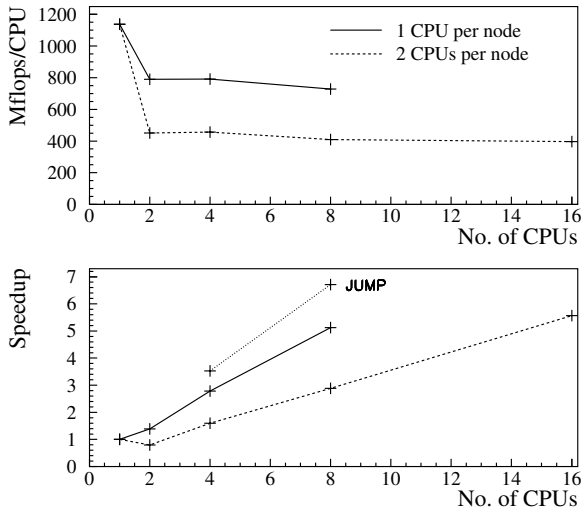

Figure 2: Speed in Mflops/CPU and speedup for a Lattice–QCD application using one or two CPUs per node. For comparison the speedup-plot also shows values taken on an IBM p690 SMP node with 32 1.7 GHz Power4+ CPUs of the Jülich JUMP cluster. These numbers were kindly provided by [10].

RFIO OVER INFINIBAND

RFIO

Remote File Input/Output (RFIO) is an efficient method to access files on remote hosts. The development started 1990 at CERN as part of the SHIFT project. Today, RFIO is part of the CASTOR[11] storage manager project which provides transparent tape access using the RFIO protocol. RFIO consists of a set of command line tools, e.g. rfcp, rfdir or rfrm, an API library providing POSIX like functions to access the files on the remote host, such as rfio_open(), rfio_read() etc., and the rfiod daemon, the server program. Remote file access can be random access or in a streaming mode.

Using InfiniBand with RFIO

Goal of the InfiniBand port of RFIO (RFIO/IB) was to allow fast file transfers at very low CPU utilization. This is possible by combining the 'reliable connection' and

RDMA features of InfiniBand. A simple RDMA based single stream transfer protocol has been implemented as a replacement of the standard streaming protocol as used by rfcp. The connection to the server is established via Ethernet. The use of the InfiniBand fabric requires the exchange of additional information between server and client e.g. connection end points, buffer addresses and access keys for RDMA write transfers. The port is based on the Mellanox Verbs API (VAPI)[12]. The current Linux HCA drivers require the server process to be fully multi-threaded. Every time the server receives a connection request, a new thread is created to take care of the data transfers. The software is available for download as a patch against the most recent version of CASTOR, 1.7.1.5. More details and first performance results of a pair of Xeon test nodes were published in [13].

Comparison of Xeon and Opteron systems

Since [13] RFIO/IB has been improved in terms of stability and support for 64-bit architectures, like the AMD Opteron. Figures 3 and 4 show a comparison of the read performance for unfilled buffers and real file transfers between Xeon-Xeon (solid) and Opteron-Opteron (dashed). By transferring unfilled buffers, the pure network+protocol bandwidth can be measured since no time is required to fill meaningful data into the transfer buffers. The better memory performance of the Opteron architecture is reflected in the significantly larger throughput seen on those systems, while the CPU utilization remains at a low level on both architectures, particularly for large files. The relatively high CPU consumption for the transfer of small amounts of data is caused by the initialization of the connection. Figure 5 shows a comparison of the remote write performance of the two systems. Again, the better memory performance of the Opteron is reflected in a significantly larger throughput. CPU utilization on the server is higher due to the filling of the RDMA transfer buffers within the read() function before the transfer.

Aggregate throughput

A typical situation occurring in high energy physics analysis environments is when many people run their analyses simultaneously on the same input data file. Physics data sets are usually written once but are read many times by many different people. Once the Large Hadron Collider (LHC) is up and running, hundreds of people will be interested to analyze the most recent data as fast and soon as possible. In a worst case scenario, they will try to access the same file on the same server from many different clients at the same time. This situation is likely to happen, and it is necessary to study the behavior of the server in such a situation.

A single quad-Opteron system with 16 GB of memory was used as a file server running the rfiod daemon. The server was equipped with a Mellanox InfiniHost 4X HCA. The operating system was SuSE SLES 8 for Opteron. 12

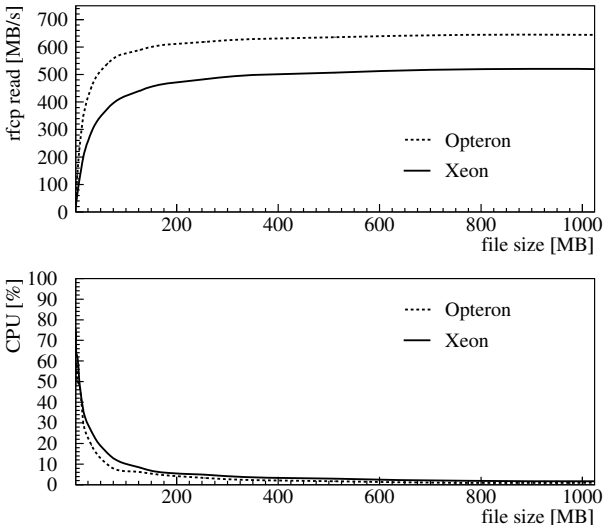

Figure 3: Comparison of network plus protocol performance and CPU usage for transfers between a pair of Opteron (dashed) and Xeon (solid) systems.

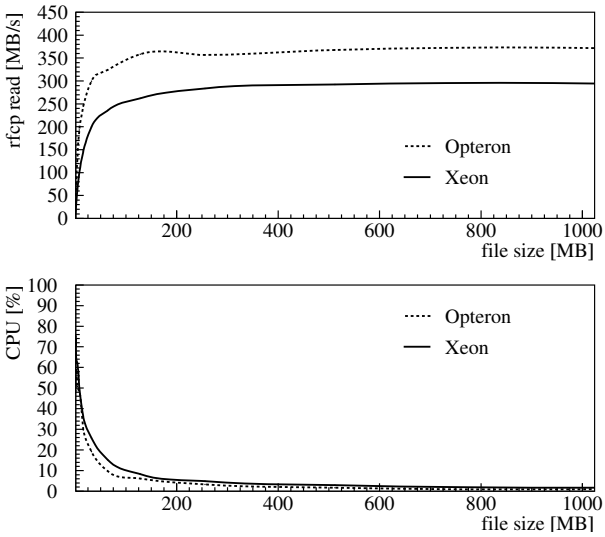

Figure 4: Comparison of the read performance and CPU usage for a cached file between a pair of Opteron (dashed) and Xeon (solid) systems, dropping the output at the client side.

dual-Xeon nodes, running the clients and the quad-Opteron file server were connected to a 16-port Mellanox 4X InfiniBand switch. Up to two client processes (readers) were running on each Xeon node. All readers read the same random data input file of 1 GB size simultaneously and drop the output to /dev/null. The test file remained cached in main memory off the server. Each client read the same file 250 times. Figure 6 shows the aggregate throughput delivered by the server as well as the average client read performance. During the measurement, the CPU utilization on the server did not exceed 150% (where 100% corresponds to 1 CPU). To a single reader, the server delivers

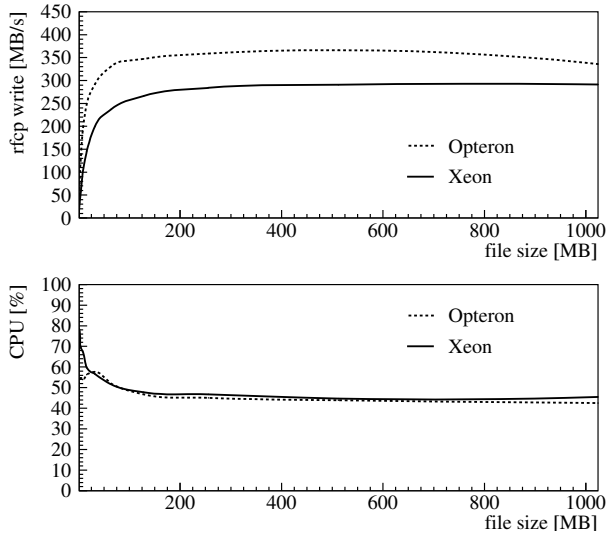

Figure 5: Comparison of the write performance and CPU usage for a cached file between a pair of Opteron (dashed) and Xeon (solid) systems, dropping the input at the server side.

approximately 350 MB/s. For more than one reader, the total throughput increases and saturates at almost 700 MB/s. Three readers saturate the link. The aggregate throughput remains stable up to the maximum available number of 24 readers. The total amount of data moved during this test was roughly 30 TB.

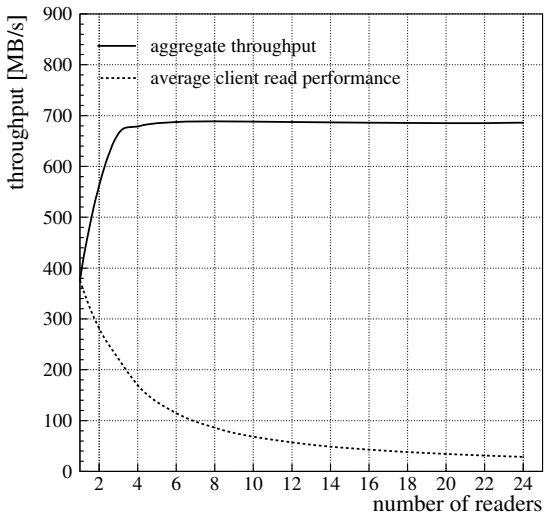

Figure 6: Measurement of the total throughput delivered by a single server machine as a function of the number of readers, all reading the same file.

SUMMARY AND OUTLOOK

Infiniband is an emerging interconnect technology with interesting features like remote DMA and reliable connections. Due to its low latency and high bandwidth it is interesting for building inexpensive clusters providing comparatively good performance for parallel computing applications. The large bandwidth and RDMA capabilities allow fast and reliable data transfers at very low CPU consumption. A single AMD-Opteron based server has been shown to be able to provide a stream of in total 700 MB/s of data to numerous clients, using a home grown port of CASTOR's RFIO protocol to InfiniBand.

In high energy physics environments, ROOT[14] is a popular object-oriented framework for interactive and non-interactive data analysis. Many LHC experiments will use it as the standard analysis system. Apart from the rfio interface, ROOT has its own protocol and server software, ROOTD as well as the newer XROOTD[15], to support remote file access. The Parallel ROOT Facility (PROOF) allows the parallel execution of analysis scripts on clusters of heterogeneous computers. Based on the experiences with RFIO a promising project is a port of the ROOT file servers to native InfiniBand. Such an interface will enable users to profit directly from the features of the interconnect in a transparent way, particularly in a clustered framework using PROOF.

REFERENCES

[1] InfiniBand Trade Association, InfiniBand Architecture Specification, Release 1.1, November 6th, 2002.

[2] D.K. Panda, MPI over InfiniBand Project. Network-Based Computing Laboratory, Ohio-State University, http://nowlab.cis.ohio-state.edu/projects/mpi-iba

[3] Scali MPI connect, http://www.scali.com;

[4] NCSA VMI, http://vmi.ncsa.uiuc.edu

[5] Mellanox Technologies, http://www.mellanox.com

[6] http://www.openib.org

[7] http://linux.web.cern.ch/linux/scientific3

[8] http://www-unix.mcs.anl.gov/mpi/mpich

[9] A. Petitet, R. C. Whaley, J. Dongarra, A. Cleary, High-Performance Linpack benchmark, http://www.netlib.org/benchmark/hpl

[10] Carsten Urbach, FU Berlin, Fachbereich Physik, priv. com.

[11] CASTOR (CERN Advanced Storage Manager), http://castor.web.cern.ch

[12] Mellanox Technologies, Mellanox IB-Verbs API (VAPI), Rev. 1.00, February 2004.

[13] 'First Experience with the InfiniBand Interconnect', Ulrich Schwickerath, Andreas Heiss, Proceedings of ACAT03, December 1-5, 2003, accepted for publication in Nuclear Inst. and Methods in Physics Research, A

[14] The ROOT System Homepage, http://root.cern.ch

[15] The XROOTD project homepage can be found at http://xrootd.slac.stanford.edu